

ElektroUbayaSot – Simulator Robot Soccer Menggunakan Webots

Haryadi Poedji Sudrajat

Teknik Elektro – Universitas Surabaya

e-mail: hary_ps.ubaya@yahoo.com

Abstrak – Simulator robot *soccer* ini dibuat untuk digunakan pada ajang pelatihan pemrograman Elektro UBAYA. Pada ajang pelatihan sebelumnya menggunakan simulator Simurosot dan masalah yang muncul adalah masih diperlukan program tambahan berupa *notepad* serta masih menggunakan simulator buatan mahasiswa universitas lain. Pada simulator Simurosot, simulasi dijalankan dengan tampilan 3 dimensi. Simulator ElektroUbayaSot dibuat menggunakan program Webots, sehingga simulasi yang ditampilkan dapat berbentuk 3 dimensi dan dapat dijalankan layaknya pertandingan sesungguhnya. Hasil yang didapat dari pengujian simulator ElektroUbayaSot adalah 4 fitur simulator ElektroUbayaSot lebih unggul dibanding fitur simulator Simurosot, waktu pertandingan simulator ElektroUbayaSot memiliki tingkat akurasi yang lebih tinggi dibanding waktu pertandingan pada simulator Simurosot dengan nilai akurasi 96.667% per 30 detik. Selain itu didapatkan bahwa tutorial yang disertakan telah lengkap dan jelas. Program *supervisor* mengawasi batas gol dengan sangat akurat pada koordinat 0.001 dan 1.299

Kata kunci: simulator, robot *soccer*, FIRA, Simurosot, Narosot, Webots.

Abstract – *This robot soccer simulator is made to be use for Elektro UBAYA programming training event. Simulator Simurosot was used at the previous programming training event and the problem that emerged that is still need a notepad and still using a program that made by other university student. Simulator Simurosot view the simulation in 3 dimensions view. Simulator ElektroUbayaSot made using Webots, so the simulation view will be in 3 dimensions view and play the simulation such as the real robot soccer match. The result is 4 features in simulator ElektroUbayaSot are more powerfull than the simulator Simurosot features. The simulator ElektroUbayaSot game timing has higher accuracy at 96.667% for every 30 seconds. It also found that all the tutorials that icluded for this simulator is complete and definite. The supervisor program supervise the goal line very accurate at coordinate point 0.001 and 1.299*

Keywords: *simulator, robot soccer, FIRA, Simurosot, Narosot, Webots.*

1. PENDAHULUAN

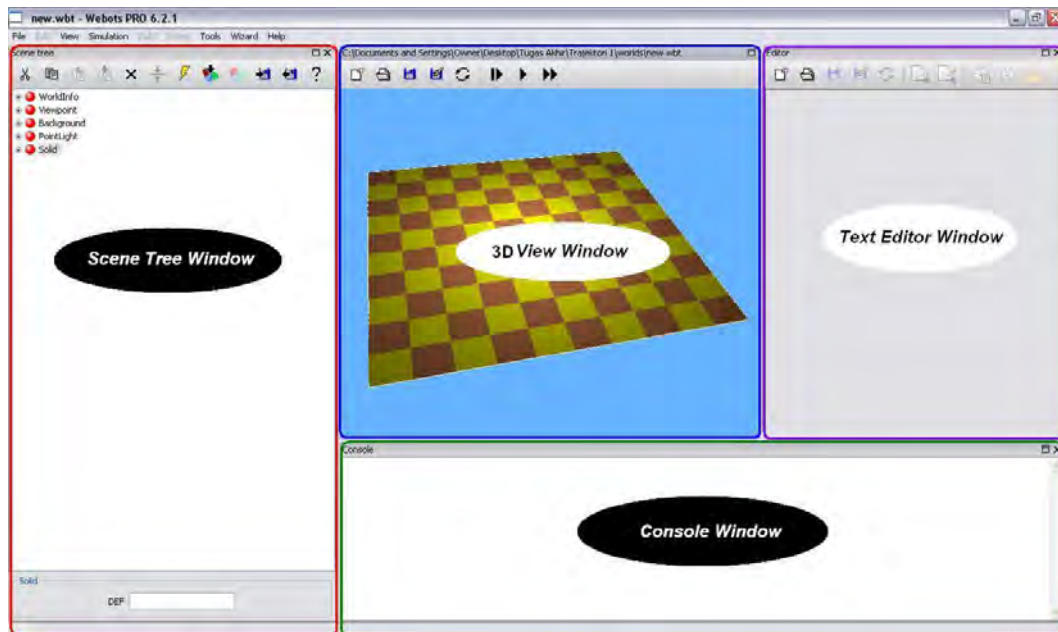
Robot *soccer* adalah pertandingan sepak bola yang tiap pemain pada pertandingan diwakilkan dengan sebuah robot beroda. Karena begitu menariknya pertandingan robot ini pada masyarakat, akhirnya dibuatlah simulator robot *soccer* oleh Roger Smith, Annabel Rodwell, Mathew Clarke, dan Shawn Truesdell serta Dr. Jun Jo dari Universitas Griffith Australia. Dengan adanya program simulator

robot *soccer* ini, maka setiap orang dapat ikut dalam pertandingan tersebut dengan lebih mudah. Simulator buatan mahasiswa-mahasiswa Griffith ini dinamakan Simurosot, dimana program ini sudah cukup lama dipublikasikan. Simurosot ini merupakan simulator robot *soccer* yang mudah digunakan dan diajarkan pada siswa sekolah menengah untuk pelatihan pemrograman yang diadakan Elektro UBAYA.

Pada penggunaanya dalam *event* pelatihan Elektro UBAYA, simulator Simurosot ini masih terdapat kekurangan terhadap fitur yang dibutuhkan saat pelatihan dilakukan. Fitur tersebut yaitu penggunaan program notepad untuk menyimpan data-data tim perwakilan tiap sekolah (nama tim dan nama sekolah). Di setiap UBAYA RoboGames diadakan, Elektro UBAYA masih memakai simulator Simurosot buatan mahasiswa-mahasiswa Universitas Griffith tersebut untuk pertandingan simulator robot *soccer*. Selain itu simulator Simurosot ini adalah program yang bebas diunduh oleh siapa saja, sehingga bisa saja universitas lain melakukan ajang pelatihan dengan simulator yang sama sehingga Elektro UBAYA tidak memiliki nilai lebih.

FIRA mempunyai banyak *event* pertandingan robot *soccer* yaitu Hurosot, Amiresot, Miroso, Androsot, Robosot, Narosot, dan Simurosot. Pada pengerjaan Tugas Akhir ini, peraturan-peraturan, dan spesifikasi-spesifikasi robot yang dibuat mengusung peraturan-peraturan, dan spesifikasi-spesifikasi seperti pada pertandingan Narosot. Fitur-fitur yang dibuat pada Tugas Akhir ini mengacu pada fitur-fitur perantara, dan fitur-fitur pendukung pada simulator Simurosot dan beberapa fitur tambahan yang dibutuhkan. Karena pada Tugas Akhir ini diinginkan sebuah simulator robot *soccer* yang dapat berjalan seperti pertandingan sesungguhnya sehingga dapat mengadopsi/membayangkan seperti *event* pertandingan sesungguhnya (real/bukan simulator) maka simulator robot *soccer* yang dibuat pada Tugas Akhir ini dibuat dengan simulator robot yaitu Webots.

Pada Tugas Akhir ini direalisasikan sebuah simulator robot *soccer* yang memiliki spesifikasi-spesifikasi seperti pertandingan Narosot, fitur-fitur seperti pada simulator Simurosot dengan 2 fitur tambahan sesuai keperluan pada ajang pelatihan pemrograman Elektro UBAYA serta simulator mampu menampilkan pertandingan layaknya pertandingan sesungguhnya.



Gambar 1. Tampilan Awal Webots

2. DASAR TEORI

Webots adalah aplikasi simulasi robot profesional yang dapat membuat dunia *virtual* dengan berbagai faktor yang dapat mempengaruhinya seperti gravitasi, gesekan, dan sebagainya. Webots menyediakan berbagai macam *sensor* dan aktuator seperti *servo*, *sensor* jarak, *driver* motor, kamera dan lain-lain. Pada Webots, *user* dapat memprogram robot menggunakan bahasa pemrograman C, C++, Java, Phyon, dan MATLAB™. Pada Tugas Akhir ini bahasa pemrograman yang digunakan adalah bahasa pemrograman C [1].

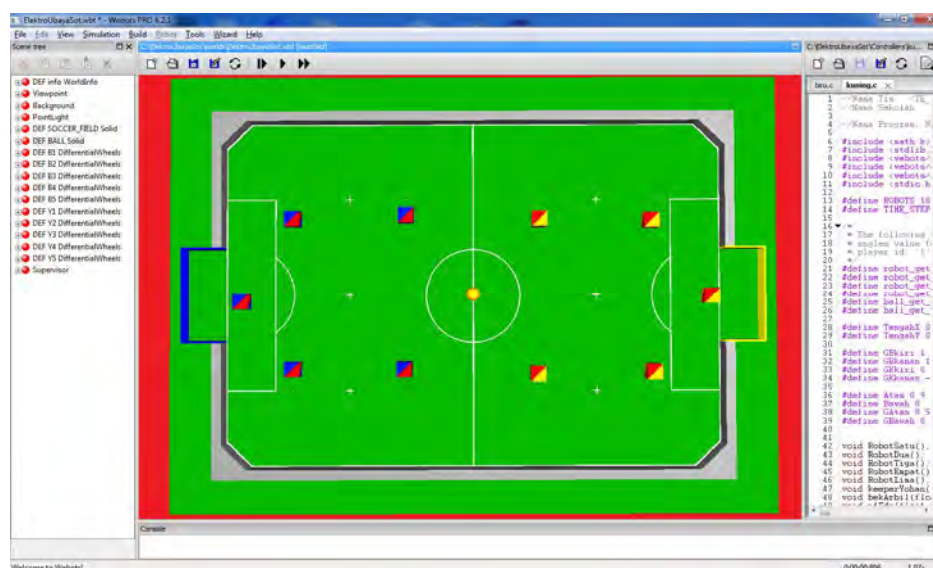
Tampilan awal pada Webots terbagi atas 4 jendela yaitu *scene tree window*, *3D view window*, *text editor window*, dan *console window* yang ditunjukkan pada Gambar 1. *Scene tree window* menampilkan informasi-informasi dan representasi hirarki dari objek yang ditampilkan pada *3D view window* seperti robot dan lingkungan simulasi. Pada *3D view window* itu sendiri, *3D view window* merupakan jendela yang menampilkan simulasi robot dan pada *3D view window* ini *user* dapat berinteraksi dengan robot pada saat simulasi dijalankan. *Text editor window* memberikan fasilitas kepada *user* untuk menuliskan/modifikasi *source code* sesuai keinginan dan *user* dapat melakukan kompilasi setelah *source code* selesai dibuat. *Console window* menampilkan hasil kompilasi dari *source code* berupa *warning* atau *error*, serta menampilkan *output* dari *controller* [1].

3. DESAIN

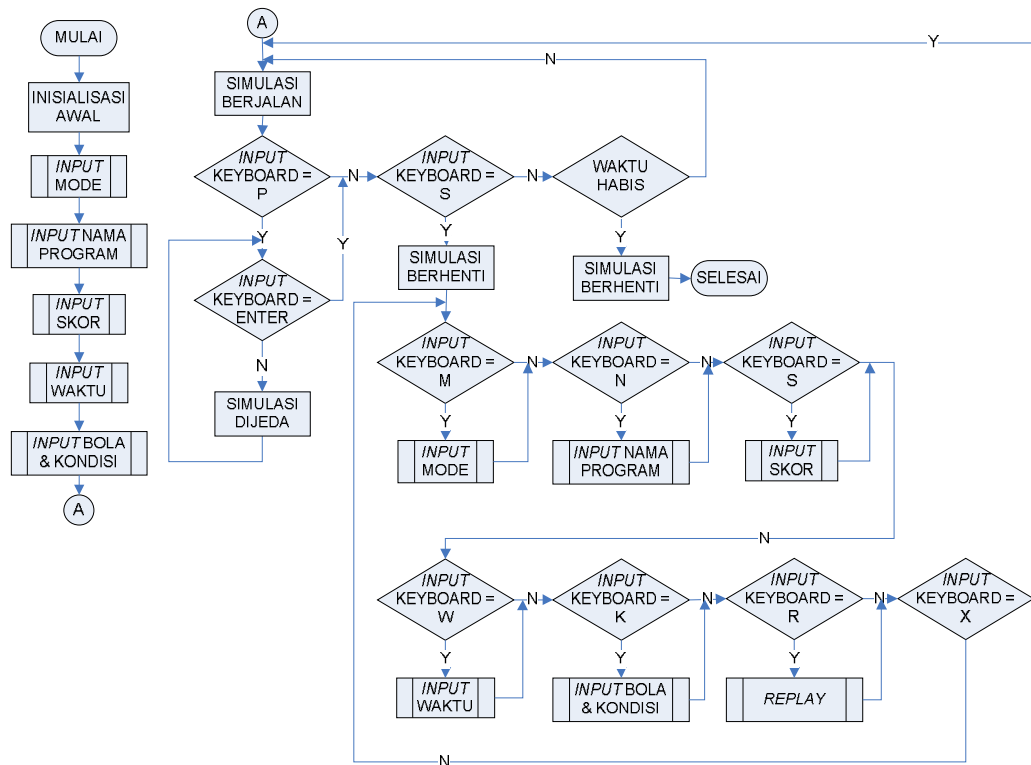
Pada simulator Webots, telah disertakan rancangan desain robot *soccer* yang dinamakan “soccer.wbt”. Desain simulator ElektroUbayaSot adalah dengan memodifikasi rancangan “soccer.wbt” dan langkah-langkahnya adalah membuat tampilan alas lapangan, memodifikasi dinding lapangan, memodifikasi gawang, memodifikasi robot, memodifikasi bola, memodifikasi program *supervisor*, memodifikasi program robot *user*, dan membuat tutorial. Pada rancangan “soccer.wbt” jumlah robot tiap tim adalah 3 robot pemain. Dapat dilihat pada Gambar 2 setelah membuat alas lapangan, memodifikasi alas lapangan, memodifikasi dinding lapangan, memodifikasi gawang, memodifikasi robot, dan memodifikasi bola terbentuklah desain fisik keseluruhan dari ElektroUbayaSot.

Pada desain fitur, langkah yang dilakukan adalah memodifikasi program *supervisor* yang telah disertakan. Fitur yang didesain adalah fitur-fitur perantara dan fitur-fitur pendukung pertandingan. Dapat dilihat pada Gambar 3 adalah flowchart dari program pengendali fitur dan setelah modifikasi selesai dilakukan, terbentuklah fitur-fitur pada ElektroUbayaSot yang ditunjukkan pada Gambar 4.

Pada desain program robot *user*, langkah yang dilakukan adalah memodifikasi program robot *user* yang telah disertakan. Program dimodifikasi sedemikian sehingga dapat bergerak dengan memiliki karakter masing-masing pada tiap robot. Dapat dilihat pada Gambar 5 adalah bagan isi dari program robot *user* simulator ElektroUbayaSot.



Gambar 2. Bentuk Keseluruhan Simulator ElektroUbayaSot



Gambar 3. Flowchart Program Pengendali Fitur



Gambar 4. Fitur pada Simulator ElektroUbayaSot

```

1 //Nama Tim : <TE_Ubaya>
2 //Nama Sekolah : <Ubaya>
...
4 //Nama Program, Nama Tim & Nama sekolah masing-masing maks. 15 karakter
5
...
6 #include <math.h>
7 #include <stdlib.h>
8 #include <webots/robot.h>
9 #include <webots/differential_wheels.h>
10 #include <webots/receiver.h>
11 #include <stdio.h>
12
13 #define ROBOTS 10
14 #define TIME STEP 64
...
240 wb_differential_wheels_set_speed(left_speed, right_speed);
241 }
242
243 wb_robot_cleanup();
244
245 return 0;
246 }
247
248 /***** USER ADVANCE *****/
...
248 /***** USER ADVANCE *****/
249
250 /***** KEEPER *****/
251 /***** KEEPER *****/
252 /***** KEEPER *****/
253 void keeperYohan(float x1, float x2, float y1, float y2)
254 {
255     vl = 0;
256     vr = 0;
257     if(x < x1)
258
...
452 /***** END USER ADVANCE *****/
453
454 void RobotSatu()
455 {
456     /***** USER *****/
457     keeperYohan(0.001, 0.123, GBawah, GAtas);
458



```

Gambar 5. Bagan Isi Program Robot *User*

4. PENGUJIAN DAN HASIL PENGUJIAN

Pada tahap pengujian dilakukan, pertama-tama dilakukan pengujian kelengkapan fitur. Untuk mengetahui kelengkapan fitur yang ada pada simulator ElektroUbayaSot ini, fitur pada simulator ElektroUbayaSot dibandingkan dengan fitur pada simulator Simurosot. Dari data yang ditunjukkan pada **Error! Not a valid bookmark self-reference.**, diketahui bahwa fitur pada simulator ElektroUbayaSot lebih lengkap dan terdapat 4 fitur pada simulator ElektroUbayaSot lebih unggul dibanding fitur pada simulator Simurosot dan 3 fitur pada simulator Simurosot lebih unggul dibanding fitur pada simulator ElektroUbayaSot.

Tabel 1. Pengujian Kelengkapan Fitur Simulator ElektroUbayaSot

		
Simulator	ElektroUbayaSot	Simurosot
Fitur		
<i>Input Mode</i>	Ada, 2 Langkah	Tidak Ada
<i>Input Nama Program</i>	Ada, 3 Langkah	Ada, 6 Langkah
<i>Input Nama Tim dan Input Nama Sekolah</i>	Ada, 2 Langkah	Tidak Ada
<i>Input Skor</i>	Ada, 3 Langkah	Ada, 3 Langkah
<i>Input Waktu</i>	Ada, 3 Langkah	Ada, 4 Langkah atau Lebih
<i>Input Kondisi Bola</i>	Ada, 3 Langkah	Ada, 2 Langkah
<i>Input Kondisi Pertandingan</i>	Ada, 3 Langkah	Ada, 1 Langkah
<i>Input Jeda, Berhenti, dan Replay Pertandingan</i>	Ada. Jeda dan Berhenti = 1 Langkah Replay = 2 Langkah	Ada, 1 Langkah
<i>Tampilan Koordinat Posisi Tiap Robot</i>	Ada, 3 Langkah	Ada, 3 Langkah

Setelah pengujian kelengkapan fitur telah dilakukan, selanjutnya adalah melakukan pengujian waktu pertandingan. Pada pengujian waktu pertandingan ini dilakukan dengan 3 komputer yang memiliki spesifikasi berbeda-beda. Dapat dilihat pada Tabel 2 hasil pengujian EUS yang menunjukkan akurasi waktu pertandingan pada simulator ElektroUbayaSot dan SS yang menunjukkan akurasi waktu pertandingan pada simulator Simurosot. Dapat dilihat pada Gambar 6 adalah hasil pengujian ke-3 pada komputer ke-2 dari waktu pertandingan pada simulator ElektroUbayaSot memiliki toleransi 1 detik tiap 30 detik. Sehingga

didapat tingkat akurasi waktu pertandingan simulator ElektroUbayaSot dibanding waktu yang sesungguhnya adalah 96.667% tiap 30 detik. Dapat dilihat juga pada Gambar 7 adalah hasil pengujian ke-3 pada komputer ke-2 dari waktu pertandingan pada simulator Simurosot yang memiliki toleransi mencapai 5 detik tiap 30 detik. Sehingga diketahui akurasi waktu pertandingan pada simulator Simurosot adalah 83.333% tiap 30 detik.

Tabel 2. Pengujian Waktu Pertandingan

Pengujian ke -	Komputer 1 AMD Athlon(tm) 64 Processor 3500+, MMX, 3D Now @2.2 GHz 896 MB RAM NVIDIA GeForce 6100, 512 MB		Komputer 2 Intel(R) Core(TM) i7-2635QM @2 GHz (8 CPUs) 4096 MB RAM AMD RADEON HD 6490M, 256 MB		Komputer 3 Intel(R) Core(TM) i3-2310M @2.1 GHz (4 CPUs) 2048 MB RAM NVIDIA GeForce GT 520M, 1024 MB	
	EUS	SS	EUS	SS	EUS	SS
1	100%	84.444%	98.333%	76.667%	100%	93.333%
2	99.444%	80%	98.333%	80%	100%	70%
3	98.333%	81.667%	96.667%	83.333%	99.444%	73.333%
4	98.333%	86.667%	98.889%	80%	98.889%	76.667%
5	98.333%	86.667%	98.889%	76.667%	98.889%	70%



Gambar 6. Pengujian Waktu Pertandingan Simulator ElektroUbayaSot

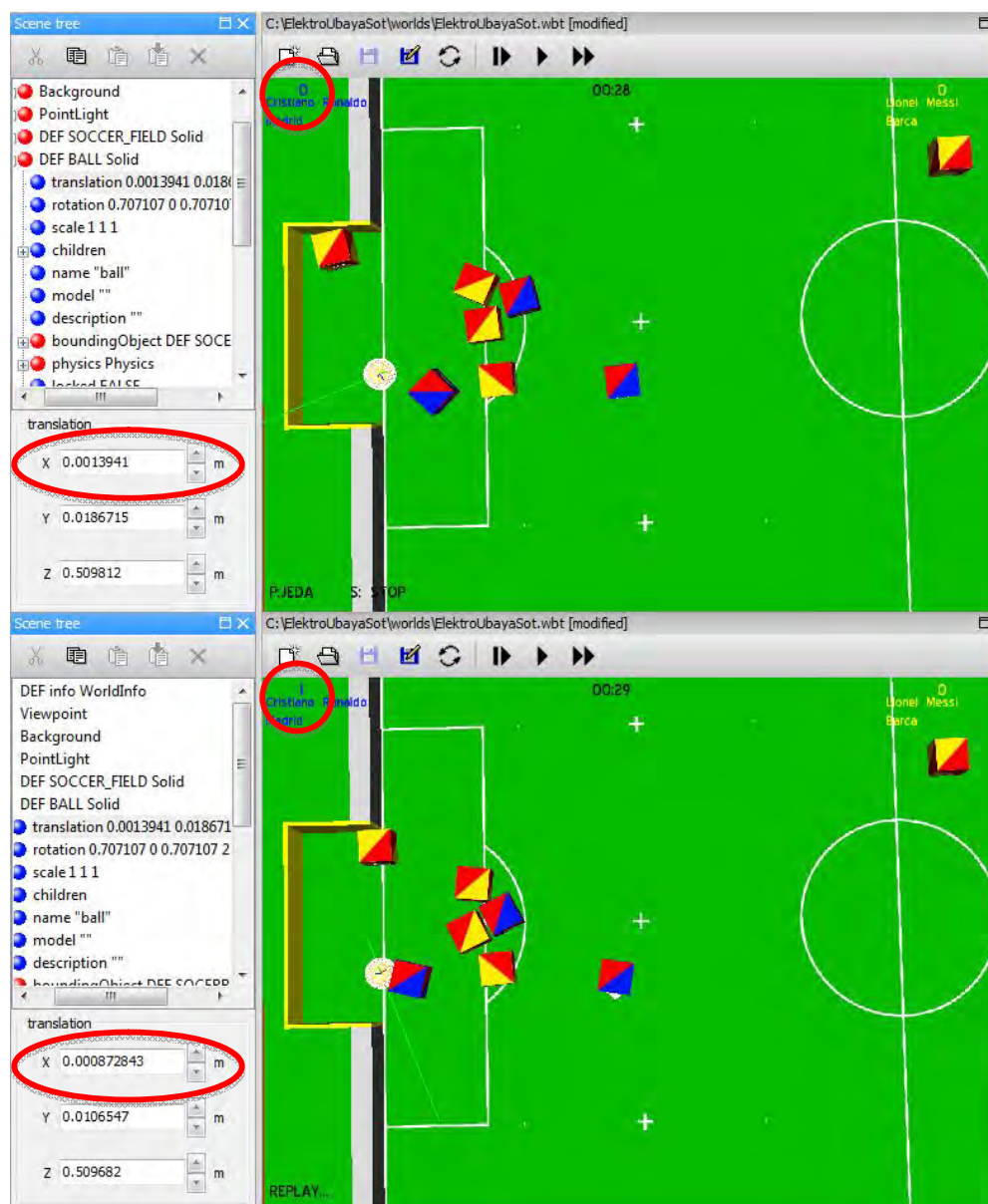


Gambar 7. Pengujian Waktu Pertandingan Simulator Simurosot

Pada pengujian penambahan skor saat terjadi gol, dapat dilihat pada Gambar 8 bahwa program *supervisor* mengawasi batas garis area gol dengan tingkat akurasi yang tinggi, Gambar 8 menunjukkan saat tim biru menyerang gawang tim kuning dan dapat diketahui bahwa program *supervisor* menentukan terjadi sebuah gol saat titik tengah bola belum benar-benar melewati garis area gol pada koordinat 0.001.

Pada pengujian tutorial simulator ElektroUbayaSot ini terdapat 2 tahap pengujian yaitu tahap pengujian oleh asisten dosen dan tahap pengujian secara global oleh populasi mahasiswa kelas robotika. Pengujian ke-1 dilakukan dengan tujuan agar mendapatkan timbal balik lebih cepat, sehingga apabila terdapat kekurangan dapat segera direvisi. Dapat dilihat pada Tabel 3 bahwa hasil pengujian tutorial ke-1 menunjukkan bahwa tutorial sudah lengkap dan cukup jelas. Pada tutorial “*Compile Menggunakan Visual C*”, asisten dosen mengajukan beberapa pertanyaan sederhana.

Pada pengujian tutorial ke-2 yang dilakukan oleh 14 peserta mahasiswa kelas robotika diketahui bahwa *compile* program menggunakan visual C dapat menyebabkan simulasi yang dijalankan ElektroUbayaSot menjadi jauh lebih lambat, sehingga tutorial “*Compile Menggunakan Visual C*” dihapuskan. Pada pengujian tutorial ke-2 juga didapatkan beberapa pertanyaan tentang kemenangan pertandingan yang tidak dapat ditentukan dengan adu penalti dan dari pertimbangan yang dilakukan setelah adu penalti tidak dapat menentukan kemenangannya, maka dilakukan pertandingan ulang tanpa ada peraturan.



Gambar 8. Pengujian Penambahan Skor Saat Terjadi Gol

Tabel 3. Pengujian Tutorial Ke-1

Tutorial	Responden	Pertanyaan
Hal Utama	Budiyanto D.	Tidak ada
Instalasi Webots	Budiyanto D.	Tidak ada
Instalasi Visual C	Budiyanto D.	Tidak ada
Aturan ElektroUbayaSot	Budiyanto D.	Tidak ada
Nama Robot	Budiyanto D.	Tidak ada
Koordinat Lapangan	Budiyanto D.	Tidak ada
<i>Compile</i> menggunakan Visual C	Budiyanto D.	Ada
Modifikasi Sederhana	Budiyanto D.	Tidak ada
Membuat Program Sederhana	Budiyanto D.	Tidak ada

Tabel 4. Pengujian Tutorial Ke-2

No	Item	Ya	Tidak
1	Kebutuhan Menjalankan ElektroUbayaSot		√
2	Instalasi Webots		√
3	Instalasi Visual C		√
4	Aturan Pertandingan ElektroUbayaSot		
a	Sistem Pertandingan		√
b	Lama Pertandingan		√
c	Memulai Pertandingan		√
d	Kemenangan	√	
e	Pelanggaran		√
f	Tendangan Bebas		√
g	Tendangan Gawang		√
h	Tendangan Penalti	√	
i	Bola Bebas		√
j	Permohonan Robot Terbalik		√
k	Bola Mati		√
5	Nama Robot		√
6	Koordinat Lapangan		√
7	Compile Program Menggunakan Visual C	√	
8	Modifikasi Sederhana		
a	Mengganti Formasi Pemain		√
b	Modifikasi Letak Pemain	√	
c	Menentukan Bola		√
d	Menentukan Flag		√
9	Membuat Program Sederhana		
a	Menggerakkan Robot		√
b	Membuat Penjaga Gawang Mengikuti Koordinat Y Dari Bola		√
c	Membuat Penjaga Gawang Tetap Lurus Ke Arah 3.14 Radian		√

d	Membatasi Gerak Penjaga Gawang		√
e	Membuat Prosedur Penjaga Gawang		√
f	Membuat Bek		√
g	Membuat Gelandang Yang Terus Mengejar Bola		√
h	Tendangan Penalti		√
i	Bola Bebas		√

Tabel 5. *Format* Kuisioner Pengujian Kemudahan Penggunaan Fitur

PERTANYAAN	NILAI				
	1	2	3	4	5
Menggeser robot.	1	2	3	4	5
Memutar rotasi robot.	1	2	3	4	5
Memulai/menjalankan pertandingan.	1	2	3	4	5
Memasukkan nama tim dan nama sekolah.	1	2	3	4	5
Memasukkan nama program.	1	2	3	4	5
Pengaktifan tampilan nomor punggung robot.	1	2	3	4	5
Mengganti nilai skor.	1	2	3	4	5
Mengganti nilai waktu.	1	2	3	4	5
Memasukkan kondisi bola dan kondisi pertandingan.	1	2	3	4	5
Menjeda pertandingan.	1	2	3	4	5
Menghentikan pertandingan.	1	2	3	4	5
<i>Replay</i> pertandingan.	1	2	3	4	5
Menampilkan posisi tiap robot dan bola.	1	2	3	4	5

Selain itu, pada pengujian tutorial ke-2 juga ditanyakan mengenai penanda agar robot dapat mengetahui posisi tim yang membawa bola, dan kondisi pertandingan. Dari pertimbangan yang didapat, keterangan tentang penanda kondisi bola dan kondisi pertandingan dijelaskan/ditambahkan di dalam tutorial “Modifikasi Sederhana”. Beberapa peserta juga menanyakan tentang cara memindahkan dan memutar robot yang jatuh, bagian ini memang belum dijelaskan di tutorial manapun dan dari pertimbangan yang didapat keterangan mengenai memindahkan dan memutar robot yang jatuh dijelaskan pada 1 tutorial tambahan yang menjadi tutorial ke-9 yaitu tutorial “Panduan Wasit”.

Pada pengujian kemudahan penggunaan fitur, dilakukan dengan 5 responden yang diminta untuk mengoperasikan fitur-fitur pada simulator ElektroUbayaSot lalu mengisi kuisioner seperti yang ditunjukkan pada Tabel 5. Pengujian dilakukan 3 hari dengan hari pertama adalah pengoperasian fitur berdasarkan penjelasan dari pelatih dan tutorial, pada hari kedua hanya dengan tutorial, dan pada hari ketiga mandiri tanpa penjelasan pelatih maupun tutorial. Hasil dari pengujian didapat bahwa pengoperasian fitur pada simulator ElektroUbayaSot ini tergolong mudah untuk dioperasikan.

Pada pengujian *minimum hardware requirement* diasumsikan bahwa spesifikasi minimum komputer yang dapat digunakan adalah komputer yang mampu menjalankan simulator ElektroUbayaSot dengan kecepatan simulasi 6x dari kecepatan simulasi yang sesungguhnya. Dapat dilihat pada Tabel 6 dan Tabel 7 bahwa pengujian *minimum hardware requirement* ini dilakukan dengan 8 komputer yang memiliki spesifikasi berbeda-beda dan didapatkan bahwa spesifikasi komputer minimum yang digunakan adalah prosesor yang setara dengan Intel(R) Core(TM)2 Quad Q8400 @2.66 GHz (4 CPUs), RAM = 1 GB, VGA setara 384 MB (*Total Memory*), dan komputer hanya menjalankan simulator ElektroUbayaSot.

Tabel 6. Tabel Pengujian Kecepatan Simulasi (1)

Komputer	AMD Athlon(tm) 64 Processor 3500+, MMX, 3D Now @2.2 GHz 896 MB RAM NVIDIA GeForce 6100, 512 MB	Intel(R) Core(TM) i7-2635QM @2 GHz (8 CPUs) 4096 MB RAM AMD RADEON HD 6490M, 256 MB	Intel(R) Core(TM) i3-2310M @2.1 GHz (4 CPUs) 2048 MB RAM NVIDIA GeForce GT 520M, 1024 MB	Intel(R) Core(TM) i5-3317U @1.7 GHz (4 CPUs) 8192 MB RAM Intel(R) HD Graphics 4000, 64 MB
Kecepatan	4x	6x atau lebih	6x atau lebih	6x atau lebih

Tabel 7. Tabel Pengujian Kecepatan Simulasi (2)

Komputer	Intel(R) Core(TM) i3 @2.13 GHz (4 CPUs) 3072 MB RAM ATI Mobility Radeon HD5145, 512 MB	Intel(R) Core(TM) i5 @2.53 GHz (4 CPUs) 2048 MB RAM ATI Mobility Radeon HD5470, 512 MB	Intel(R) Core(TM) i3 @2.4 GHz (4 CPUs) 4096 MB RAM Intel(R) HD Graphics, 64 MB	Intel(R) Core(TM)2 Quad Q8400 @2.66 GHz (4 CPUs) 1014 MB RAM Intel(R) G33/G31 Express Chipset Family, 384 MB
Kecepatan	6x atau lebih	6x atau lebih	6x atau lebih	6x atau lebih

Pada simulator ElektroUbayaSot, simulasi yang dijalankan dapat disimulasikan seperti pertandingan sesungguhnya. Sehingga robot pada simulator ElektroUbayaSot ini dapat jatuh pada kecepatan tinggi. Pada pengujian kecepatan maksimum robot ini didapatkan kecepatan maksimum robot agar dapat berjalan stabil adalah pada kecepatan 20 rad/s.

5. SIMPULAN dan SARAN

Simulator ElektroUbayaSot berhasil berjalan dengan baik dengan 8 fitur yang ada seperti pada simulator Simurosot dengan 2 fitur tambahan yang disediakan khusus pada simulator ElektroUbayaSot. Fitur *input* mode memudahkan pengguna simulator ElektroUbayaSot saat melakukan pemrograman. Dengan fitur-fitur yang ada pada simulator ElektroUbayaSot didapat 4 fitur lebih unggul dan 3 fitur lebih lemah dibanding fitur-fitur pada simulator Simurosot.

Akurasi waktu pertandingan pada simulator ElektroUbayaSot lebih tinggi dibanding akurasi waktu pertandingan pada simulator Simurosot dengan nilai akurasi 96.667%. Akurasi program *supervisor* dalam mengawasi bola dan garis area gol sangat tinggi dalam menentukan terjadinya gol. Tutorial yang disertakan pada simulator ElektroUbayaSot ini telah lengkap dan jelas. Simulator ElektroUbayaSot berhasil dijalankan dengan kecepatan 6x kecepatan simulasi sesungguhnya dengan spesifikasi *minimum hardware requirement* adalah prosesor

yang setara dengan Intel(R) Core(TM)2 Quad Q8400 @2.66 GHz (4 CPUs), RAM = 1 GB, VGA setara 384 MB (*Total Memory*), dan komputer hanya menjalankan simulator ElektroUbayaSot.

Saran untuk pengembangan dan perbaikan terhadap simulator ElektroUbayaSot ini adalah membuat bentuk badan robot yang memiliki keseimbangan lebih baik agar robot dapat digerakkan dengan kecepatan lebih dari 20 rad/s, membuat waktu pertandingan yang mengikuti perhitungan *interrupt timer*, memodifikasi fitur *input keyboard* agar lebih responsif/stabil di berbagai komputer atau memodifikasi *input* fitur-fitur dengan tampilan tombol berbasis JAVA.

DAFTAR PUSTAKA

- [1] Oey Donna, D.S. 2012. *Optimasi Teknik Pengeboran PCB Menggunakan Fuzzy C-means pada Simulator Webots*. Tugas Akhir. Surabaya: Universitas Surabaya.
- [2] Webots *Reference Manual release 6.2.1*. Cyberbotics Ltd, 2010.
- [3] The Federation of International Robot-soccer Association (FIRA), 1998. Narosot. *Aturan Pertandingan Narosot FIRA*. (Online), (<http://fira.net/?mid=narosot>).
- [4] Smith, R., Rodwell, A., Clarke, M., & Truesdell, S. 2006. *3D Robot Soccer Simulator Simurosot*. (Online), (<http://fira.net/?mid=simurosot>, diakses 11 Agustus 2011).